Remarks:

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-15, 17-20, 22-32 are pending in the application. No claims have been allowed. Claims 1, 12, 13, 17, 24, 27 and 28 are independent. Claims 1, 12, 13, 17, 24, 26, 27, and 28 have been amended.

Cited Art

The Office Action cites U.S. Patent No. 5,586,304 to Stupek, Jr. et al. ("Stupek"), U.S. Patent No. 5,493,682 to Tyra et al. ("Tyra"), and U.S. Patent No. 6,802,061 to Parthasarathy ("Parthasarathy").

§ 103 Rejections

The Action rejected claims 1-9, 11-15, 17-20, 22-29, and 32 under 35 U.S.C. § 103(a) as unpatentable over Stupek in view of Tyra. Applicants respectfully submit the claims in their present form are allowable over the cited art.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP § 2142. Motivations to combine or modify references must come from the references themselves or be within the body of knowledge in the art. MPEP § 2143.01.

Claims 1, 12, 13, 17, 24, 27 and 28

Claims 1, 12, 13, 17, 24, 27 and 28 have been amended to clarify how dependencies are acquired. Amended claim 1 reads as follows (emphasis added):

> responsive to determining the software associated with the software dependency is not present on the computer, *acquiring, by a software package manager running on the computer*, the software associated with the software dependency;

Amended claim 12 reads as follows (emphasis added):

> responsive to determining the software associated with the software dependency is not present on the computer, *acquiring, by a software package*

> *manager running on the computer*, the software associated with the software dependency; and

Amended claim 13 reads as follows (emphasis added):

> responsive to determining the version installed at the computer is not sufficient, *acquiring* and installing, *by a software package manager running on the computer*, the software dependency, wherein the acquiring and installing the software dependency comprises recursively specifying the one or more other software dependencies.

Amended claim 17 reads as follows (emphasis added):

> responsive to determining software dependencies associated with the specified software are not already installed at the computer, *acquiring, by the software package manager running on the computer*, the software dependencies;

Amended claim 24 reads as follows (emphasis added):

> wherein the nestable software package format is operable to specify a remote location from which one or more components required by the software package can be *acquired by a software package manager running on a computer*, and the nestable software package format is operable to be recursively processed by recursively processing the list of dependencies before finishing installation of the software package.

Amended claim 27 reads as follows (emphasis added):

> a dependency list indicating one or more items depended on by the software, wherein at least one of the items on the dependency list is not contained in the package, and the software package indicates a remote location from which the item can be *acquired* and installed *by a software package manager running on the computer*;

Amended claim 28 reads as follows (emphasis added):

> a *software package manager running on the computer system* operable to resolve the specified list of one or more dependencies by consulting the database to determine whether a dependency is installed and further operable to *acquire* a dependency determined as not installed;

For example, the Application at page 4, line 7 to page 5, line 3, describes acquiring dependencies by the package manager as follows:

> A software package manager uses a distribution unit containing components for a software package and a manifest file that describes the distribution unit to manage the installation, execution, and uninstallation of software packages on a computer. For installation, the package manager acquires the manifest file and parses it to learn if the software package depends on any

> additional components. The package manager resolves any dependencies by acquiring a distribution unit containing the needed component and installs the dependency's distribution unit as described below. Because dependencies can nested within dependencies, the package manager recursively processes all the dependencies before finishing the installation of the software package that depends upon the additional components.
>
> The software package manager acquires the distribution unit and extracts the components in the distribution unit into a directory on the computer. The package manager causes the operating system of the computer to install the software. The package manager then updates a code store data structure with information in the manifest file. The fields in the code store data structure contains such information as the name and version of the distribution unit, a list of the components and their location on the computer, and the source of the distribution unit. Additional fields in the code store data structure can also contain a component version, a component data type, and a digital signature if one was affixed to the distribution unit.

Additional description of acquiring dependencies by the software package manager can be found, for example, in the Application at page 14, lines 6-15.

Stupek's description of dependencies does not teach or suggest acquiring dependencies by a software package manager running on a computer, as described by the above-cited language of claims 1, 12, 13, 17, 24, 27 and 28. Regarding dependencies, Stupek describes that:

> The database also contains information regarding the dependencies between the package and other upgrade objects or packages: ...
>
> While each upgrade distribution medium will commonly contain all upgrade packages upon which a particular upgrade depends, it is also likely that upgrades to a package will depend upon upgraded packages not stored on the distribution medium. For example, the printing capabilities of an upgraded word processor created by one vendor may depend upon an upgrade to a printer driver produced by another vendor. While it is unlikely that the word processor upgrade and the driver upgrade will be distributed on the same CD, the user should still be informed of the dependency. Therefore, the dependency information in the Package database 25 describes not only the dependencies between packages on the CD, but also all dependencies between an upgrade package and any upgrade not available on the CD. Even though the unavailable upgrades cannot be automatically installed with the available upgrades, the user is nonetheless aware of their necessity. Stupek col. 6, lines 24-51.

While Stupek describes that information regarding dependencies is provided, Stupek describes that "unavailable upgrades cannot be automatically installed with the available upgrades." Accordingly, Stupek's shortfall is illustrated by Stupek's example of a word processor upgrade that is dependent upon a printer driver upgrade where the printer driver upgrade (a dependency) cannot be automatically installed if it is not on the distribution medium along with the word processor upgrade. Regarding dependencies that "cannot be automatically installed," Stupek describes that the user is informed that they exist.

Applicants recognized this situation described by Stupek regarding dependencies as a problem. For example, the Application at p. 3, lines 3-8, describes the problem with obtaining dependencies as follows:

> Although most current software is written in modules, there is no current mechanism that handles the situation where one component in a software program requires the presence of another to operate. If a user downloads software from a Web page, the user may discover that the program requires an external library which necessitates another network session to download, assuming the user can find the right location, and then the user must manually install the library before installing the software.

Applicants solve this problem by acquiring dependencies by a software package manager running on a computer.

Because Stupek describes that dependencies that are not on the distribution medium "cannot be automatically installed," Stupek does not teach or suggest "responsive to determining the software associated with the software dependency is not present on the computer, acquiring, by a software package manager running on the computer, the software associated with the software dependency" as recited by claim 1, "responsive to determining the software associated with the software dependency is not present on the computer, acquiring, by a software package manager running on the computer, the software associated with the software dependency" as recited by claim 12, "responsive to determining the version installed at the computer is not sufficient, acquiring and installing, by a software package manager running on the computer, the software dependency, wherein the acquiring and installing the software dependency comprises recursively specifying the one or more other software dependencies" as recited by claim 13, "responsive to determining software dependencies associated with the specified software are not already installed at the computer, acquiring, by the software package manager running on the computer, the software dependencies" as recited by claim 17, "wherein the nestable software package format is operable to specify a remote location from which one or more components required by the software package can be acquired by a software package manager running on a computer, and the nestable software package format is operable to be recursively processed by recursively processing the list of dependencies before finishing installation of the software package" as recited by claim 24, "a dependency list indicating one or more items depended on by the software, wherein at least one of the items on the dependency list is not contained in the package, and the software package indicates a remote location from which the item can be acquired and installed by a software package manager running on the computer" as recited by claim 27, and "a software package manager running on the computer system operable to resolve the specified list of one or more dependencies by consulting the database to determine whether a dependency is installed and further operable to acquire a dependency determined as not installed" as recited by claim 28.

Furthermore, as understood by Applicants, Tyra, separately or in combination with Stupek, does not teach or suggest the above-cited language of claims 1, 12, 13, 17, 24, 27 and 28. Therefore, claims 1, 12, 13, 17, 24, 27 and 28 should be in condition for allowance.

Claims 2-9, 11, 14, 15, 18-20, 22, 23, 25, 26, 29, and 32

Claims 2-9, 11, and 32 ultimately depend on claim 1. Thus, for at least the reasons set forth above with regard to claim 1, claims 2-9, 11, and 32 should be in condition for allowance.

Claims 14 and 15 depend on claim 13. Thus, for at least the reasons set forth above with regard to claim 13, claims 14 and 15 should be in condition for allowance.

Claims 2-9 and 11 ultimately depend on claim 1. Thus, for at least the reasons set forth above with regard to claim 1, claims 2-9 and 11 should be in condition for allowance.

Claims 18-20, 22, and 23 depend on claim 17. Thus, for at least the reasons set forth above with regard to claim 17, claims 18-20, 22, and 23 should be in condition for allowance.

Dependent claim 26 has been amended to correspond with independent claim 24. Claims 25 and 26 ultimately depend on claim 24. Thus, for at least the reasons set forth above with regard to claim 24, claims 25 and 26 should be in condition for allowance.

Claim 29 depend on claim 28. Thus, for at least the reasons set forth above with regard to claim 28, claim 29 should be in condition for allowance.

The Action rejected claims 10, 30, and 31 under 35 U.S.C. § 103(a) as unpatentable over Stupek in view of Tyra and Parthasarathy. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claims 10, 30, and 31

Claim 10 ultimately depends from claim 1, and claims 30 and 31 depend from claim 28. Therefore, for at least the reasons stated above with regard to claims 1 and 28, the Action's

§ 103(a) rejection of claims 10, 30, and 31 also cannot be supported. Thus, the claims should be in condition for allowance.

Request for Interview

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office Action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

Conclusion

The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By G. L. Maurer
Gregory L. Maurer
Registration No. 43,781